WHAT IS CLAIMED IS:

1. A computer system, comprising:

an interconnect;

a plurality of processor nodes, coupled to the interconnect, each processor node including:

at least one processor core, each processor core having an associated memory cache for caching memory lines of information;

an interface to a local memory subsystem, the local memory subsystem storing a multiplicity of memory lines of information; and

a protocol engine implementing a predefined cache coherence protocol; and

a plurality of input/output nodes, coupled to the interconnect, each input/output node including:

no processor cores;

an input/output interface for interfacing to an input/output bus or input/output device;

a memory cache for caching memory lines of information;

an interface to a local memory subsystem, the local memory subsystem storing a multiplicity of memory lines of information; and

a protocol engine implementing the predefined cache coherence protocol.

2. The system of claim 1, wherein

the protocol engine of each of the processor nodes enables the processor cores therein to access memory lines of information stored in the local memory subsystem and memory lines of information stored in the memory cache of any of the processor nodes and input/output nodes, and maintains cache coherence between memory lines of information cached in the memory caches of the processor nodes and memory lines of information cached in the memory caches of the input/output nodes; and

the protocol engine of each of the input/output nodes enables an input/output device coupled to the input/output interface of the input/output node to access memory lines of information stored in the local memory subsystem and memory lines of information stored in the memory cache of any of the processor nodes and input/output nodes, and maintains cache coherence between memory lines of information cached in the memory caches of the processor nodes and memory lines of information cached in the memory caches of the input/output nodes.

3. The system of claim 1, wherein the system is reconfigurable so as to include any ratio of processor node to input/output nodes so long as a total number of processor nodes and input/output nodes does not exceed a predefined maximum number of nodes.

4. The system of claim 1, wherein the protocol engine of each of the processor nodes is functionally identical to the protocol engine of each of the input/output nodes.

5. The system of claim 1, wherein the protocol engine of each of the processor nodes and the protocol engine of each of the input/output nodes includes:

a memory transaction array for storing an entry related to a memory transaction, the entry including a memory transaction state, the memory transaction concerning a memory line of information; and

logic for processing the memory transaction, including advancing the memory transaction when predefined criteria are satisfied and storing a state of the memory transaction in the memory transaction array.

6. The system of claim 5, wherein the protocol engine of each of the processor nodes and the protocol engine of each of the input/output nodes is configured to add an entry related to a memory transaction in the memory transaction array in response to receipt by the protocol engine of a protocol message related to the memory transaction.

7. The system of claim 1, wherein

the processor nodes and the input/output nodes collectively comprise nodes of the system;

each node of the processor nodes and the input/output nodes includes:

a directory including a respective entry associated with each respective memory line of information stored in the local memory subsystem of the node, the entry including an identification field for identifying a subset of the system nodes caching the memory line of information; and

the protocol engine of each of the processor nodes and the protocol engine of each of the input/output nodes includes logic for:

configuring the identification field of each directory entry to comprise a plurality of bits at associated positions within the identification field;

associating with each respective bit of the identification field one or more nodes of the plurality of nodes, including a respective first node, wherein the one or more nodes associated with each respective bit are determined by reference to the position of the respective bit within the identification field;

setting each bit in the identification field of the directory entry associated with the memory line for which the memory line is cached in at least one of the associated nodes; and

sending an initial invalidation request to no more than a first predefined number of the nodes associated with set bits in the identification field of the directory entry associated with the memory line.

8. The system of claim 1, wherein

the processor nodes and the input/output nodes collectively comprise nodes of the system;

each node of the processor nodes and the input/output nodes includes:

input logic for receiving a first invalidation request, the invalidation request identifying a memory line of information and including a pattern of bits for identifying a subset of the plurality of nodes that potentially store cached copies of the identified memory line; and

processing circuitry, responsive to receipt of the first invalidation request, for determining a next node identified by the pattern of bits in the invalidation request and for sending to the next node, if any, a second invalidation request corresponding to the first invalidation request, and for invalidating a cached copy of the identified memory line, if any, in the particular node of the processor computer system.

9. A computer system, comprising:

a plurality of multiprocessor nodes, each multiprocessor node including:

a multiplicity of processor cores, each processor core having an associated memory cache for caching memory lines of information;

an interface to a local memory subsystem, the local memory subsystem storing a multiplicity of memory lines of information; and

a protocol engine implementing a predefined cache coherence protocol; and

a plurality of input/output nodes, each input/output node including:

no processor cores;

an input/output interface for interfacing to an input/output bus or input/output device;

a memory cache for caching memory lines of information;

an interface to a local memory subsystem, the local memory subsystem storing a multiplicity of memory lines of information; and

a protocol engine implementing the predefined cache coherence protocol.

10. The system of claim 9, wherein

the protocol engine of each of the multiprocessor nodes enables the processor cores therein to access memory lines of information stored in the local memory subsystem and memory lines of information stored in the memory cache of any of the multiprocessor nodes and input/output nodes, and maintains cache coherence between memory lines of information cached in the memory caches of the multiprocessor nodes and memory lines of information cached in the memory caches of the input/output nodes; and

the protocol engine of each of the input/output nodes enables an input/output device coupled to the input/output interface of the input/output node to access memory lines of information stored in the local memory subsystem and memory lines of information stored in the memory cache of any of the multiprocessor nodes and input/output nodes, and maintains cache coherence between memory lines of information cached in the memory caches of the multiprocessor nodes and memory lines of information cached in the memory caches of the input/output nodes.

11. The system of claim 9, wherein the system is reconfigurable so as to include any ratio of multiprocessor node to input/output nodes so long as a total number of multiprocessor nodes and input/output nodes does not exceed a predefined maximum number of nodes.

12. The system of claim 9, wherein the protocol engine of each of the multiprocessor nodes is functionally identical to the protocol engine of each of the input/output nodes.

13. The system of claim 9, wherein the protocol engine of each of the multiprocessor nodes and the protocol engine of each of the input/output nodes includes:

a memory transaction array for storing an entry related to a memory transaction, the entry including a memory transaction state, the memory transaction concerning a memory line of information; and

logic for processing the memory transaction, including advancing the memory transaction when predefined criteria are satisfied and storing a state of the memory transaction in the memory transaction array.

14. The system of claim 13, wherein the protocol engine of each of the multiprocessor nodes and the protocol engine of each of the input/output nodes is configured to add an entry related to a memory transaction in the memory transaction array in response to receipt by the protocol engine of a protocol message related to the memory transaction.

15. The system of claim 9, wherein

the multiprocessor nodes and the input/output nodes collectively comprise nodes of the system;

each node of the multiprocessor nodes and the input/output nodes includes:

a directory including a respective entry associated with each respective memory line of information stored in the local memory subsystem of the node, the entry including an identification field for identifying a subset of the system nodes caching the memory line of information; and

the protocol engine of each of the multiprocessor nodes and the protocol engine of each of the input/output nodes includes logic for:

configuring the identification field of each directory entry to comprise a plurality of bits at associated positions within the identification field;

associating with each respective bit of the identification field one or more nodes of the plurality of nodes, including a respective first node, wherein the one or more nodes associated with each respective bit are determined by reference to the position of the respective bit within the identification field;

setting each bit in the identification field of the directory entry associated with the memory line for which the memory line is cached in at least one of the associated nodes; and

sending an initial invalidation request to no more than a first predefined number of the nodes associated with set bits in the identification field of the directory entry associated with the memory line.

16. The system of claim 9, wherein

the multiprocessor nodes and the input/output nodes collectively comprise nodes of the system;

each node of the multiprocessor nodes and the input/output nodes includes:

input logic for receiving a first invalidation request, the invalidation request identifying a memory line of information and including a pattern of bits for identifying a subset of the plurality of nodes that potentially store cached copies of the identified memory line; and

processing circuitry, responsive to receipt of the first invalidation request, for determining a next node identified by the pattern of bits in the invalidation request and for sending to the next node, if any, a second invalidation request corresponding to the first invalidation request, and for invalidating a cached copy of the identified memory line, if any, in the particular node of the multiprocessor computer system.